Transactions Papers

Near Optimum Error Correcting Coding And Decoding: Turbo-Codes

Claude Berrou, *Member, IEEE*, and Alain Glavieux

***Abstract*—This paper presents a new family of convolutional codes, nicknamed turbo-codes, built from a particular concatenation of two recursive systematic codes, linked together by nonuniform interleaving. Decoding calls on iterative processing in which each component decoder takes advantage of the work of the other at the previous step, with the aid of the original concept of extrinsic information. For sufficiently large interleaving sizes, the correcting performance of turbo-codes, investigated by simulation, appears to be close to the theoretical limit predicted by Shannon.**

I. Introduction

CONVOLUTIONAL error correcting or channel coding has become widespread in the design of digital transmission systems. One major reason for this is the possibility of achieving real-time decoding without noticeable information losses thanks to the well-known soft-input Viterbi algorithm [1]. Moreover, the same decoder may serve for various coding rates by means of puncturing [2], allowing the same silicon product to be used in different applications. Two kinds of convolutional codes are of practical interest: nonsystematic convolutional (NSC) and recursive systematic convolutional (RSC) codes. Though RSC codes have the same free distance d_f as NSC codes and exhibit better performance at low signal to noise ratios (SNR's) and/or when punctured, only NSC codes have actually been considered for channel coding, except in Trellis-coded modulation (TCM) [3]. Section II presents the principle and the performance of RSC codes, which are at the root of the study expounded in this article.

For a given rate, the error-correcting power of convolutional codes, measured as the coding gain at a certain binary error rate (BER) in comparison with the uncoded transmission, grows more or less linearly with code memory ν. Fig. 1 (from [4]) shows the achievable coding gains for different rates, and corresponding bandwidth expansion rates, by using classical NSC codes with $\nu = 2, 4, 6$ and 8, for a BER of 10^{-6}. For instance, with $R = 1/2$, each unit added to ν adds about 0.5

Paper approved by D. Divsalar, the Editor for Coding Theory and Applications of the IEEE Communications Society. Manuscript received March . 1995; revised December 6, 1995 and February 26, 1996. This paper was presented at ICC'93, Geneva, Switzerland, May 1993.

The authors are with Ecole Nationale Supérieure des Télécommunications de Bretagne, BP 832 29285 BREST CEDEX, France.

Publisher Item Identifier S 0090-6778(96)07373-4.

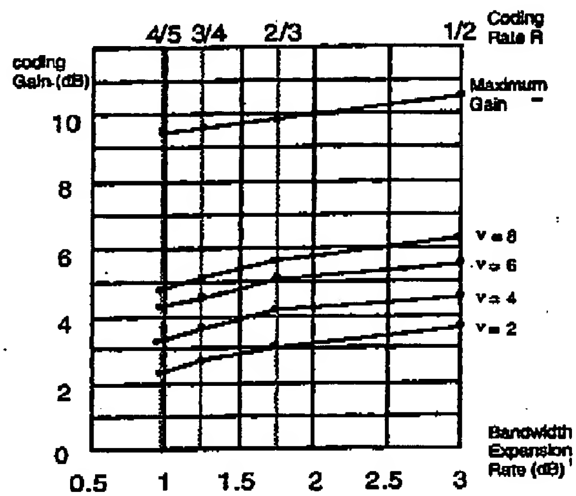


Fig. 1. Coding gains at BER equal to 10^{-6}, achievable with NSC codes (three-bit quantization, from [4], and maximum possible gains for 1/2, 2/3, 3/4, and 4/5 rates in a Gaussian channel, with quaternary phase shift keying (QPSK) modulation.

dB more to the coding gain, up to $\nu = 6$; for $\nu = 8$, the additional gain is lower. Unfortunately, the complexity of the decoder is not a linear function of ν and it grows exponentially as $\nu \cdot 2^{\nu}$. Factor 2 represents the number of states processed by the decoder and the multiplying factor ν accounts for the complexity of the memory part (metrics and survivor memory). Other technical limitations like the interconnection constraint in the silicon decoder lay-out, make the value of six a practical upper limit for ν for most applications, especially for high data rates.

In order to obtain high coding gains with moderate decoding complexity, concatenation has proved to be an attractive scheme. Classically, concatenation has consisted in cascading a block code (the outer code, typically a Reed-Solomon code) and a convolutional code (the inner code) in a serial structure. Another concatenated code, which has been given the familiar name of *turbo-code*, with an original parallel organization of two RSC elementary codes, is described in Section III. Some comments about the distance properties of this composite

0090-6778/96$05.00 © 1996 IEEE

code, are propounded. When decoded by an iterative process, turbo-codes offer near optimum performance. The way to achieve this decoding with the *Maximum A Posteriori* (MAP) algorithm is detailed in Sections IV, V, VI, and some basic results are given in Section VII.

II. Recursive Systematic Convolutional Codes

A. Introduction

Consider a binary rate $R = 1/2$ convolutional encoder with constraint length K and memory $\nu = K - 1$. The input to the encoder at time k is a bit d_k and the corresponding binary couple (X_k, Y_k) is equal to

$$X_k = \sum_{i=0}^{\nu} g_{1i} d_{k-i} \qquad g_{1i} = 0, 1 \tag{1a}$$

$$Y_k = \sum_{i=0}^{\nu} g_{2i} d_{k-i} \qquad g_{2i} = 0, 1 \tag{1b}$$

where $G_1: \{g_{1i}\}, G_2: \{g_{2i}\}$ are the two encoder generators, expressed in octal form. It is well known that the BER of a classical NSC code is lower than that of a classical nonrecursive systematic convolutional code with the same memory ν at large SNR's, since its free distance is smaller than that of a NSC code [5]. At low SNR's, it is in general the other way round. The RSC code, presented below, combines the properties of NSC and systematic codes. In particular, it can be better than the equivalent NSC code, at any SNR, for code rates larger than 2/3.

A binary rate RSC code is obtained from a NSC code by using a feedback loop and setting one of the two outputs X_k or Y_k equal to the input bit d_k. The shift register (memory) input is no longer the bit d_k but is a new binary variable a_k. If $X_k = d_k$ (respectively, $Y_k = d_k$), the output Y_k (resp. X_k) is defined by (1b) [respectively, (1a)] by substituting d_k for a_k and variable a_k is recursively calculated as

$$a_k = d_k + \sum_{i=1}^{\nu} \gamma_i a_{k-i} \tag{2}$$

where γ_i is respectively equal to g_{1i} if $X_k = d_k$ and to g_{2i} if $Y_k = d_k$. Equation (2) can be rewritten as

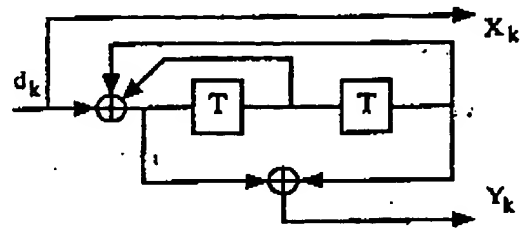
$$d_k = \sum_{i=0}^{\nu} \gamma_i a_{k-i}. \tag{3}$$

Taking into account $X_k = d_k$ or $Y_k = d_k$, the RSC encoder output $C_k = (X_k, Y_k)$ has exactly the same expression as the NSC encoder outputs if $g_{10} = g_{20} = 1$ and by substituting d_k for a_k in (1a) or (1b).

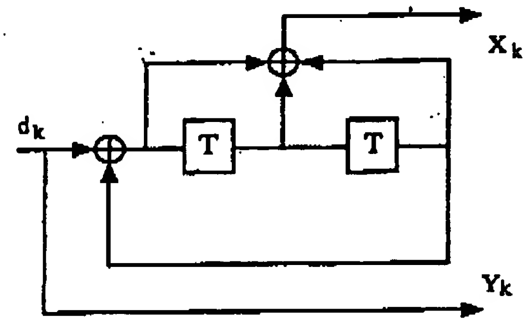
Two RSC encoders with memory $\nu = 2$ and rate $R = 1/2$, obtained from a NSC encoder defined by generators $G_1 = 7, G_2 = 5$, are depicted in Fig. 2.

Generally, we assume that the input bit d_k takes values zero or one with the same probability. From (2), we can show that variable a_k exhibits the same statistical property.

$$\begin{aligned} &\Pr\{a_k = 0 / a_{k-\nu} = \varepsilon_\nu, \cdots a_{k-i} = \varepsilon_i, \cdots a_{k-1} = \varepsilon_1\} \\ &\quad = \Pr\{d_k = \varepsilon\} = \tfrac{1}{2} \end{aligned} \tag{4}$$



(a)



(b)

Fig. 2. Two associated Recursive systematic convolutional (RSC) encoders with memory $\nu = 2$, rate $R = 1/2$ and generators $G_1 = 7, G_2 = 5$.

where ε is equal to

$$\varepsilon = \sum_{i=1}^{\nu} \gamma_i \varepsilon_i; \quad \varepsilon_i = 0, 1. \tag{5}$$

Thus, the transition state probabilities $\pi(S_k = m / S_{k-1} = m')$, where $S_k = m$ and $S_{k-1} = m'$ are, respectively, the encoder state at time k and at time $(k-1)$, are identical for the equivalent RSC and NSC codes; moreover these two codes have the same free distance d_f. However, for a same input sequence $\{d_k\}$, the two output sequences $\{X_k\}$ and $\{Y_k\}$ are different for RSC and NSC codes.

When puncturing is considered, some output bits X_k or Y_k are deleted according to a chosen perforation pattern defined by a matrix P. For instance, starting from a rate $R = 1/2$ code, the matrix P of rate 2/3 punctured code can be equal to

$$P = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}. \tag{6}$$

For the punctured RSC code, bit d_k must be emitted at each time k. This is obviously done if all the elements belonging to the first or second row of matrix P are equal to one. When the best perforation pattern for a punctured NSC code is such that matrix P has null elements in the first and the second rows, the punctured recursive convolutional code is no longer systematic. In order to use the same matrix P for both RSC and NSC codes, the RSC encoder is now defined by (3), (7), and (8)

$$Y_k = \sum_{i=0}^{\nu} \lambda_i a_{k-i} \tag{7}$$

$$X_k = d_k \tag{8}$$

ere coefficients γ_i [see (3)] and λ_i are, respectively, equal g_{1i} and g_{2i} when element $p_{1j}; 1 \le j \le n$ of matrix P is al to one and to g_{2i} and g_{1i} when p_{1j} is equal to zero.

Recursive Systematic Code Performance

n order to compare the performance of RSC and NSC es, we determined their weight spectrum and their BER. weight spectrum of a code is made up of two sets coefficients $a(d)$ and $W(d)$ obtained from two series ansions related to the code transfer function $T(D,N)$ [6]

$$T(D,N)\Big|_{N=1} = \sum_{d=d_f}^{\infty} a(d)D^d \tag{9}$$

$$\frac{\partial T(D,N)}{\partial N}\Big|_{N=1} = \sum_{d=d_f}^{\infty} W(d)D^d \tag{10}$$

re d_f is the free distance of the code, $a(d)$ is the number aths at Hamming distance d from the "null" path and) is the total Hamming weight of input sequences $\{d_k\}$ to generate all paths at distance d from the "null" path. eneral, it is not easy to calculate the transfer function punctured code, that is why the first coefficients $a(d)$ $W(d)$ are directly obtained from the trellis by using an ithm derived from [7]. From coefficients $W(d)$, a tight bound of error probability can be calculated for large 's [6]

$$P_e \le \sum_{d=d_f}^{\infty} W(d)P(d). \tag{11}$$

memoryless Gaussian channel with binary modulation , QPSK), probability $P(d)$ is equal to

$$P(d) = \frac{1}{2}\operatorname{erfc}\left[\sqrt{dR\frac{E_b}{N_0}}\right] \tag{12}$$

E_b/N_0 is the energy per information bit to noise power al density ratio and R is the code rate.

8], a large number of RSC codes have been investigated eir performance was compared to that of NSC codes, n of weight spectrum and of BER. Coefficients $a(d)$ same for RSC and NSC codes but the coefficients $(d)\}$ of RSC codes have a tendency to increase more in function of d than the coefficients $\{W_{NSC}(d)\}$ of odes, whatever the rate R and whatever the memory s, at low SNR's, the BER of the RSC code is always than the BER of the equivalent NSC code.

eneral, for rates $R \le 2/3$, the first coefficients $(d_f)), W_{RSC}(d_f+1))$ are larger than those of NSC therefore, at large SNR's, the performance of NSC s a little better than that of RSC codes. When the larger than 2/3, it is easy to find RSC codes whose ance is better than that of NSC codes at any SNR.

der to illustrate the performance of RSC codes, the RSC and NSC codes are plotted in Fig. 3 for different of R and for an encoder with memory $\nu = 6$ and rs 133, 171. For instance, at $P_e = 10^{-1}$, the coding h this RSC code, relative to the equivalent NSC code,

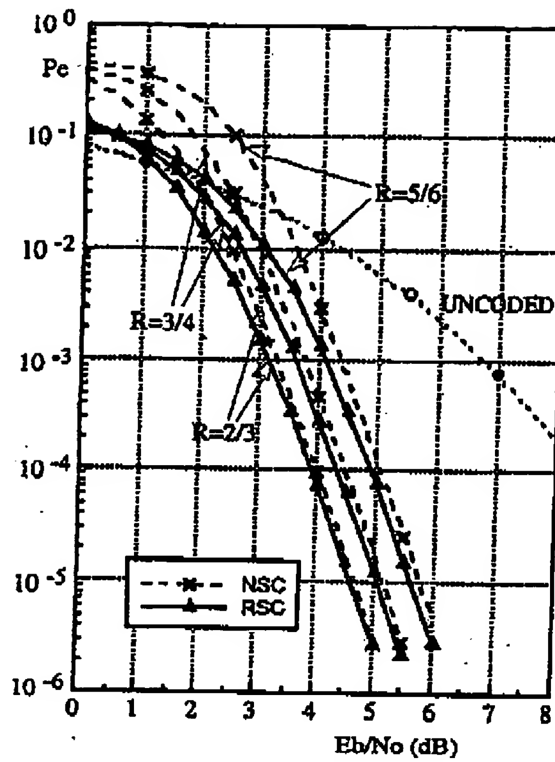


Fig. 3. P_e of punctured RSC and NSC codes for different values of rate R and memory $\nu = 6$, generators $G_1 = 133, G_2 = 171$.

is approximately 0.8 dB at $R = 2/3$, whereas at $R = 3/4$, it reaches 1.75 dB.

III. Parallel Concatenation with Non Uniform Interleaving: Turbo-Code

A. Construction of the Code

The use of systematic codes enables the construction of a concatenated encoder in the form given in Fig. 4, called *parallel concatenation*. The data flow (d_k at time k) goes directly to a first elementary RSC encoder C_1 and after interleaving, it feeds (d_n at time k) a second elementary RSC encoder C_2. These two encoders are not necessarily identical. Data d_k is systematically transmitted as symbol X_k and redundancies Y_{1k} and Y_{2k} produced by C_1 and C_2 may be completely transmitted for an $R = 1/3$ encoding or punctured for higher rates. The two elementary coding rates R_1 and R_2 associated with C_1 and C_2, after puncturing, may be different, but for the best decoding performance, they will satisfy $R_1 \le R_2$. The global rate R of the composite code, R_1 and R_2 are linked by

$$\frac{1}{R} = \frac{1}{R_1} + \frac{1}{R_2} - 1. \tag{13}$$

Unlike the classical (serial) concatenation, parallel concatenation enables the elementary encoders, and therefore the associated elementary decoders, to run with the same clock. This point constitutes an important simplification for the design of the associated circuits, in a concatenated scheme.

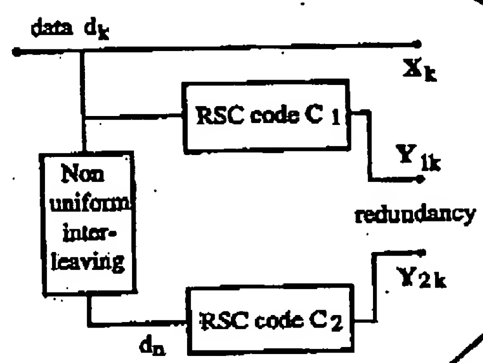


Fig. 4. Basic turbo-encoder (rate 1/3).

B. Distance Properties

Consider for instance elementary codes C_1 and C_2 with memory $\nu = 4$ and encoder polynomials $G_1 = 23, G_2 = 35$. Redundancy Y_k is one time every second time, either Y_{1k} or Y_{2k}. Then the global rate is $R = 1/2$ with $R_1 = R_2 = 2/3$. The code being linear, the distance properties will be considered relatively to the "all zero" or "null" sequence. Both encoders C_1, C_2 and the interleaver are initialized to the "all zero" state and a sequence $\{d_k\}$ containing w "1"s, is fed to the turbo-encoder. w is called the *input weight.*

Some definitions: let us call a *Finite Codeword* (FC) of an elementary RSC encoder an output sequence with a finite distance from the "all zero" sequence (i.e., a limited number of "1"s in output sequences $\{X_k\}$ and $\{Y_k\}$). Because of the linear recursivity, only some input sequences, fitting with the linear feedback register (LFR) generation of Y_k, give FC's. These particular input sequences are named *FC (input) patterns.* Let us also call a *global FC* an output sequence of the turbo-code, with a finite distance from the "all zero" sequence (i.e., a limited number of "1"s in output sequences $\{X_k\}, \{Y_{1k}\}$ and $\{Y_{2k}\}$). The distance $d_q(w)$ of an elementary FC ($q = 1$ for $C_1, q = 2$ for C_2), associated with an input sequence $\{d_k\}$ with weight w, is the sum of the two contributions of $\{X_k\}$ and $\{Y_{qk}\}$

$$d_q(w) = d_{X_q}(w) + d_{Y_q}(w). \tag{14}$$

Since the codes are systematic, $d_{X_q}(w) = w$

$$d_q(w) = w + d_{Y_q}(w). \tag{15}$$

The distance $d(w)$ of a global FC is given likewise by

$$d(w) = w + d_{Y1}(w) + d_{Y2}(w). \tag{16}$$

1) Uniform Interleaving: Consider a uniform block interleaving using an $M \cdot M$ square matrix with M large enough (i.e., ≥ 32), and generally a power of 2. Data are written linewise and read columnwise. As explained above, the matrix is filled with "0"s except for some "1"s and we are now going to state some of the possible patterns of "1"s leading to global FC's and evaluate their associated distances. Beforehand, note that, for each elementary code, the minimal value for input weight w is two, because of their recursive structure. For the particular case of $w = 2$, the delay between the two data at "1" is 15 or a multiple of 15, since the LFR associated with the redundancy generation, with polynomial $G = 23$, is a maximum length LFR. If the delay is not a multiple of 15, then the associated sequence $\{Y_k\}$ will contain "0"s and "1"s indefinitely.

Global FC's with Input Weight $w = 2$

Global FC's with $w = 2$ have to be FC's with input weight $w = 2$ for each of the constituent codes. Then the two data at "1" in the interleaving memory must be located at places which are distant by a multiple of 15, when both writing and reading. For lack of an extensive analysis of the possible patterns which satisfy this double constraint, let us consider only the case where the two "1"s in $\{d_k\}$ are time-separated by the lowest value (i. e. 15). It is also a FC pattern for code C_2 if the two data at "1" are memorized on a single line, when writing in the interleaving memory, and thus the span between the two "1"s is $15 \cdot M$, when reading. From (16), the distance associated with this input pattern is approximately equal to

$$\begin{aligned} d(2) &\approx 2 + \min\{d_{Y1}(2)\} + \mathrm{INT}((15 \cdot M + 1)/4) \\ &= 2 + 4 + \mathrm{INT}((15 \cdot M + 1)/4) \end{aligned} \tag{17}$$

where INT$(\cdot)$ stands for integer part of $(\cdot)$. The first term represents input weight w, the second term the distance added by redundancy Y_1 of C_1 (rate 2/3), the third term the distance added by redundancy Y_2. The latter is given assuming that, for this second 2/3 rate code, the $(15 \cdot M + 1)/2$ Y_2 symbols are at "1", one out of two times statistically, which explains the additional division by 2. With $M = 64$ for instance, the distance is $d(2) \approx 246$. This value is for a particular case of a pattern of two "1"s but we imagine it is a realistic example for all FC's with input weight 2. If the two "1"s are not located on a single line when writing in the interleaving matrix, and if M is larger than 30, the span between the two "1"s, when reading from the interleaving matrix, is higher than $15 \cdot M$ and the last term in (17) is increased.

Global FC's with $w = 3$

Global FC's with input weight $w = 3$ have to be elementary FC's with input weight $w = 3$ for each of the constituent codes. The inventory of the patterns with three "1"s which satisfy this double constraint is not easy to make. It is not easier to give the slightest example. Nevertheless, we can consider that associated distances are similar to the case of input weight two codewords, because the FC for C_2 is still several times M long.

Global FC's with $w = 4$

Here is the first pattern of a global FC which can be viewed as the separate combination of two minimal (input weight $w = 2$) elementary FC patterns both for C_1 and C_2. When writing in the interleaving memory, the four data at "1" are located at the four corners of a square or a rectangle, the sides of which have lengths equal to 15 or a multiple of 15 [Fig. 5(a)]. The minimum distance is given by a square pattern, with side length equal to 15, corresponding to minimum values